Support Material

GCE Computing

OCR Advanced GCE in Computing: H447

Unit: F453

This Support Material booklet is designed to accompany the OCR Advanced GCE specification in Computing for teaching from September 2008.

© OCR 2007

Contents

Contents **2**

Introduction **3**

Scheme of Work **5**

Lesson Plan **17**

Other forms of Support 25

Introduction

Background

A new structure of assessment for A Level has been introduced, for first teaching from September 2008. Some of the changes include:

- The introduction of stretch and challenge (including the new A* grade at A2) – to ensure that every young person has the opportunity to reach their full potential
- The reduction or removal of coursework components for many qualifications – to lessen the volume of marking for teachers
- A reduction in the number of units for many qualifications – to lessen the amount of assessment for learners
- Amendments to the content of specifications – to ensure that content is up-to-date and relevant.

OCR has produced an overview document, which summarises the changes to Computing. This can be found at **www.ocr.org.uk,** along with the new specification.

In order to help you plan effectively for the implementation of the new specification we have produced this Scheme of Work and Sample Lesson Plans for Computing. These Support Materials are designed for guidance only and play a secondary role to the Specification.

Our Ethos

All our Support Materials were produced 'by teachers for teachers' in order to capture real life current teaching practices and they are based around OCR's revised specifications. The aim is for the support materials to inspire teachers and facilitate different ideas and teaching practices.

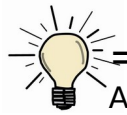
Each Scheme of Work and set of sample Lesson Plans is provided in:

- PDF format – for immediate use
- Word format – so that you can use it as a foundation to build upon and amend the content to suit your teaching style and students' needs.

The Scheme of Work and sample Lesson plans provide examples of how to teach this unit and the teaching hours are suggestions only. Some or all of it may be applicable to your teaching.

The Specification is the document on which assessment is based and specifies what content and skills need to be covered in delivering the course. At all times, therefore, this Support Material booklet should be read in conjunction with the Specification. If clarification on a particular point is sought then that clarification should be found in the Specification itself.

A Guided Tour through the Scheme of Work



= Innovative Teaching Idea

All the teaching idea contained in the SOW are innovative, but the icon is used to Highlight exceptionally innovative ideas.

= Stretch & Challenge Activity

This icon is added at the end of text when there is an explicit opportunity to offer
Stretch and Challenge.

= ICT Opportunity

This icon is used to illustrate when an activity could be taught using ICT facilities.

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.1 THE FUNCTION OF OPERATING SYSTEMS

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Features of operating systems		a describe the main features of operating systems – for example, memory management, scheduling algorithms;		
• Interrupt handling		b explain how interrupts are used to obtain processor time and how processing of interrupted jobs may later be resumed, (typical sources of interrupts should be identified and any algorithms and data structures should be described);	• include an awareness that an interrupt may have a lower priority than the current task	
• Scheduling, job queues and priorities		c define and explain the purpose of scheduling, job queues, priorities and how they are used to manage job throughput;	• note that a job's priority may change to ensure it is processed	
• Memory management		d explain how memory is managed in a typical modern computer system, (virtual memory, paging and segmentation should be described along with some of the	• physical and logical division of memory and programs • (avoid the common error of confusing memory with disk	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.1 THE FUNCTION OF OPERATING SYSTEMS

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
		problems which could occur such as disk threshing	storage)	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.1 THE FUNCTION OF OPERATING SYSTEMS

Suggested teaching time range		**Specification content**	Teacher guidance

SECTION SUMMARY	**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Spooling	e describe spooling, explaining why it is used;	• temporary storage to avoid delays due to speed mismatch	
• Modern personal computer operating systems	f describe the main components of a typical desktop PC operating system, including the file allocation table (FAT) and how it is used, and the purpose of the boot file.	• avoid confusion between FAT and disk directory (as for item d. earlier in this section)	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.2 THE FUNCTION AND PURPOSE OF TRANSLATORS

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Types of translators and their use		a. describe the need for and use of translators to convert source code to object code; b. understand the relationship between assembly language and machine code; c. describe the use of an assembler in producing machine code; d. describe the difference between interpretation and compilation; e. describe the purpose of intermediate code in a virtual machine;	• an overview: types of code, translation between them, error detection and reporting	
• Lexical analysis		f. describe what happens during lexical analysis;		
• Syntax analysis		g. describe what happens during syntax analysis, explaining how errors are handled;	• include examples of errors (not just spelling mistakes)	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.2 THE FUNCTION AND PURPOSE OF TRANSLATORS

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Code generation and optimisation		h. explain the code generation phase and understand the need for optimisation	• include ability to optimise for size or speed	
• Library routines		i. describe the use of library routines.	• understand the need for library routines and how they work with a program	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.3 COMPUTER ARCHITECTURES

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Von Neumann architecture • Registers – purpose and use		a. describe classic Von Neumann architecture, identifying the need for, and the uses of, special registers in the functioning of a processor	• idea of a register, with examples to include PC, MAR, MDR, CIR	
• Fetch-execute cycle		b. **describe, in simple terms, the fetch/decode/execute cycle, and the effects of the stages of the cycle on specific registers;**	• as above	
• Other machine architectures		c. discuss co-processor, parallel processor and array processor systems, their uses, advantages and disadvantages;	• include comparisons of use for specific types of task (but omit relative cost)	
		d. describe and distinguish between Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC) architectures.	• features of each, with relative advantages/disadvantages	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.4 DATA REPRESENTATION

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Floating point binary		a. demonstrate an understanding of floating point representation of a real binary number;	• terms should include mantissa and exponent binary numbers may be in sign and magnitude or two's complement notation	
• Normalisation of floating point binary numbers		b. normalise a real binary number;	• include the reason for normalisation • be able to normalise positive or negative numbers	
		c. discuss the trade-off between accuracy and range when representing numbers.	• relate to mantissa and exponent	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.5 DATA STRUCTURES AND DATA MANIPULATION

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Implementation of data structures, including stacks, queues and trees		a. explain how static data structures may be used to implement dynamic data structures;		
		b. describe algorithms for the insertion, retrieval and deletion of data items stored in stack, queue and tree structures;	• use algorithm for particular examples • explain features of algorithms e.g. check for stack overflow	
• Searching, merging and sorting		c. explain the difference between binary searching and serial searching, highlighting the advantages and disadvantages of each;		

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.5 DATA STRUCTURES AND DATA MANIPULATION

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
		d. explain how to merge data files;		
		e. explain the differences between the insertion and quick sort methods, highlighting the characteristics, advantages and disadvantages of each.		

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.6 HIGH LEVEL LANGUAGE PROGRAMMING PARADIGMS

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Types of languages and typical applications		a. identify a variety of programming paradigms (low level, object-oriented, declarative and procedural);		
		b. explain with examples, the terms object-oriented, declarative and procedural as applied to high level languages, showing an understanding of typical uses;	• recognise features and distinguish between types of language	
• Features of different types of language		c. discuss the concepts, and using examples, show an understanding of, data encapsulation, classes and derived classes, and inheritance when referring to object-oriented languages;		

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.6 HIGH LEVEL LANGUAGE PROGRAMMING PARADIGMS

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
		d. understand the purpose of the Unified Modelling Language (UML);		
		e. interpret class, object, use case, state, sequence, activity and communication diagrams;	• obtain information from example diagrams	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.6 HIGH LEVEL LANGUAGE PROGRAMMING PARADIGMS

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
		f. create class, object, use case and communication diagrams;	• draw diagrams or add further detail to existing diagrams	
		g. discuss the concepts, and using examples, show an understanding, of backtracking, instantiation, predicate logic and satisfying goals when referring to declarative languages.		

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.7 PROGRAMMING TECHNIQUES

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Standard programming techniques		a. explain how functions, procedures and their related variables may be used to develop a program in a structured way, using stepwise refinement;	• top down design • relate this to work done by a team of programmers	
		b. describe the use of parameters, local and global variables as standard programming techniques;	• include differences between local and global variables	
		c. explain how a stack is used to handle procedure calling and parameter passing;		
• Methods for defining syntax		d. explain the need for, and be able to create and apply, BNF (Backus-Naur form) and syntax diagrams;	• use recursion where necessary in BNF	
		e. explain the need for reverse Polish notation;		
		f. convert between reverse Polish	• use the data structures to show steps of	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.7 PROGRAMMING TECHNIQUES

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
		notation and infix form of algebraic expressions using trees and stacks.	conversion on diagrams	

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.8 LOW LEVEL LANGUAGES

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Use of computer architecture		**a. explain the concepts, and using examples, demonstrate an understanding of the use of the accumulator, registers, program counter;**	• relate this to 3.3.3	
• Features of low level languages		b. describe immediate, direct, indirect, relative and indexed addressing of memory when referring to low level languages;	• explain typical uses of types of addressing e.g. indexed addressing for items in an array	
		c. discuss the concepts and, using examples, show an understanding of mnemonics, opcode, operand and symbolic addressing in assembly language to include simple arithmetic operations, data transfer and flow-control.		

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.9 DATABASES

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Database design • Normalisation and data modelling • Methods and tools for analysing and implementing database design		a. describe flat files and relational databases, explaining the differences between them;		
		b. design a simple relational database to the third normal form (3NF), using entity-relationship (E-R) diagrams and decomposition;	• identify degree of relationship • obtain 3NF from 1NF or 2NF	
		c. define, and explain the purpose of primary, secondary and foreign keys;	• for a particular example, identify suitable keys	
• Database management system (DBMS)		d. describe the structure of a DBMS including the function and purpose of the data dictionary, data description language (DDL) and data manipulation		

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.9 DATABASES

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
		language (DML);		

GCE COMPUTING H047/H447: UNIT F453 ADVANCED COMPUTING THEORY

SECTION TITLE: 3.3.9 DATABASES

Suggested teaching time range		**Specification content**	Teacher guidance	
SECTION SUMMARY		**CANDIDATES SHOULD BE ABLE TO:**	**CLARIFICATION AND SUGGESTIONS**	**TEACHERS NOTES**
• Use of Structured Query Language (SQL)		e. use SQL to define tables and views, insert, select and delete data and to produce reports.	• understand use of views • write SQL for simple tasks • identify results of given examples in SQL	

Lesson Plan: GCE Computing H047/H447 Unit F453

3.3.3c Computer architectures

OCR recognises that the teaching of this qualification will vary greatly from school to school and from teacher to teacher. With that in mind, this lesson plan is offered as a possible approach but will be subject to modifications by the individual teacher.

Lesson length is assumed to be **one hour**.

Learning Objectives for the lesson

Objective 1	discuss co-processor, parallel processor and array processor systems, their uses, advantages and disadvantages
Objective 2	
Objective 3	

Recap of previous experience and prior knowledge

- processor components
- Von Neumann architecture

Content

Time	Content
5 minutes	Consider ways to increase speed/improve efficiency e.g. increase memory size, use cache memory, faster processor.
5-10 minutes	Discuss co-processor: increased speed, but for certain tasks only (limited use).
5-10 minutes	Discuss parallel processor: more than one processor working together, sharing same task – advantages & disadvantages
10 minutes	Discuss array processor: single processor, allows the same instruction to work on multiple data locations – advantages & disadvantages.
10-15 minutes	Give students a list of types of task. Students work in pairs/small groups to identify which type(s) of processor system is/are suitable. Feedback.

Consolidation

Time	Content
5 minutes	Recap main points & compare features.

Homework

Topic	Teacher notes
Architectures	Give further list of tasks on grid showing types of architectures: students to indicate whether task is possible, improved (e.g. faster) or not possible for each.
Exam questions	Computing 2509: June 2002 question 4 June 2007 question 3 (b)

Lesson Plan: GCE Computing H047/H447 Unit F453

3.3.6d,e,f High level language programming paradigms

OCR recognises that the teaching of this qualification will vary greatly from school to school and from teacher to teacher. With that in mind, this lesson plan is offered as a possible approach but will be subject to modifications by the individual teacher.

Lesson length is assumed to be **one hour.**

Learning objectives for the lesson

Objective 1	understand the purpose of the Unified Modelling Language (UML)
Objective 2	interpret class diagrams
Objective 3	create class diagrams

Note: further lessons will be needed for other diagrams listed in the Specification, including object and use case diagrams.

Resource needed

- An example class diagram handout prepared by the teacher – any topic relevant to students' interests e.g. vehicles

Recap of previous experience and prior knowledge

- Term "object oriented language"
- Data encapsulation, classes and derived classes, and inheritance

Content

Time	Content
5-10 minutes	List some people in school/college. Identify the classes involved e.g. Person, Student, Teacher. Discuss classes and subclasses, with attributes and methods needed to store and retrieve data. Identify some example objects.
5-10 minutes	Purpose of UML, to include need for good design, use of standard notation to avoid misunderstandings. Look at a selection of UML diagrams and discuss how they meet the purpose of UML.
10 minutes	Give out prepared class diagram. Identify features: class name, attributes, methods, arrow showing inheritance.

Time	Content
	Examples of code: discuss whether valid or not by reference to the diagram.
10 minutes	Teacher gives a different topic. Students work in pairs to produce a class diagram. Compare and discuss results.
10 minutes	Students who have had difficulty with earlier tasks should be given a further diagram to interpret, while other students work individually to produce a class diagram for another more complex topic.

Consolidation

Time	Content
5 minutes	Recap main points, with emphasis on correct notation.

Homework

Topic	Teacher notes
UML	Answer brief questions about UML.
Class diagram	Given a diagram and sample code, students should modify the diagram to show additional subclasses.
Exam questions	Computing 2509 January 2002 question 5 June 2005 question 6 (b)

Note: SAMS Teach Yourself UML in 24 Hours by Joseph Schmuller is useful for reference.

Lesson Plan: GCE Computing H047/H447 Unit F453

3.3.7d Programming techniques

OCR recognises that the teaching of this qualification will vary greatly from school to school and from teacher to teacher. With that in mind, this lesson plan is offered as a possible approach but will be subject to modifications by the individual teacher.

Lesson length is assumed to be **one hour.**

Learning objectives for the lesson

Objective 1	explain the need for, and be able to create and apply, BNF (Backus-Naur form)
Objective 2	
Objective 3	

Recap of previous experience and prior knowledge

- Writing code using variables and keywords (3.2.5)
- Syntax analysis (3.3.2)

Note: this lesson does not include syntax diagrams. Teachers may like to teach syntax diagrams first and use them to assist with BNF, particularly where recursion is involved.

Content

Time	Content
5 minutes	Introduce BNF, with the need for a notation to define terms. Standard symbols: angle brackets < > ::= is defined as | or
10 minutes	BNF definitions provided by teacher (but no recursion included). Discuss examples and decide as a group whether they conform to the definitions
10-15 minutes	Give descriptions, students work in pairs to write BNF definitions. Discuss results, checking correct use of notation.
10-15 minutes	Teacher describes an example where recursion is needed e.g. “a CODE has one letter followed by one or more digits”. Consider alternative ways to define this in BNF.

Time	Content
	Refer to recursion (mention its use in 3.2.2) and write a recursive definition for the example set.

Time	Content
10 minutes	Further examples, some with recursion, of varying difficulty according to student ability.

Consolidation

Time	Content
5 minutes	Recap main points.

Homework

Topic	Teacher notes
BNF	Examples suited to student ability: check for valid terms using definitions, or write definitions from descriptions. Include recursion for better students.
Exam questions	Computing 2509: January 2002 question 7 June 2004 question 5 (b)

Lesson Plan: GCE Computing H047/H447 Unit F453

3.3.9d Databases

OCR recognises that the teaching of this qualification will vary greatly from school to school and from teacher to teacher. With that in mind, this lesson plan is offered as a possible approach but will be subject to modifications by the individual teacher.

Lesson length is assumed to be **one hour.**

Learning objectives for the lesson

Objective 1	**function and purpose of data description language (DDL)**
Objective 2	function and purpose of data manipulation language (DML)
Objective 3	

Resources needed

- A data table design (possibly from MS Access)
- Prepared handouts with examples of DDL and DML

Recap of previous experience and prior knowledge

- Data types (3.2.3)
- Database design
- Primary and foreign keys

Content

Time	Content
10 minutes	Need for high level languages to use with databases. Using the prepared data table design, discuss tasks associated with relational databases e.g. creating tables, modifying data, producing queries.
5-10 minutes	Students list types of task, dividing them into categories for defining data or using data. Discuss.
5-10 minutes	Discuss examples of DDL with their meaning.
5-10 minutes	Discuss examples of DML with their meaning.

Time	Content
10 minutes	Students work through part of handout to explain results obtained by sample code, classifying DDL and DML. Discuss.

Consolidation

Time	Content
5 minutes	Recap and resolve any problems.

Homework

Topic	Teacher notes
DDL and DML	Students complete handout started during lesson. More able students then devise further examples to demonstrate their understanding.
Exam question	Computing 2509: January 2006 question 6 (d)

Note: To put this topic in context, it may help to refer to Computing 2509 January 2003 question 6.

Other forms of Support

In order to help you implement the new Computing specification effectively, OCR offers a comprehensive package of support. This includes:

OCR Training

Get Ready…introducing the new specifications

A series of FREE half-day training events are being run during Autumn 2007, to give you an overview of the new specifications.

Get Started…towards successful delivery of the new specifications

These full-day events will run from Spring 2008 and will look at the new specifications in more depth, with emphasis on first delivery.

Visit www.ocr.org.uk for more details.

Mill Wharf Training

Additional events are also available through our partner, Mill Wharf Training. It offers a range of courses on innovative teaching practice and whole-school issues - www.mill-wharf-training.co.uk.

e-Communities

Over 70 e-Communities offer you a fast, dynamic communication channel to make contact with other subject specialists. Our online mailing list covers a wide range of subjects and enables you to share knowledge and views via email.

Visit https://community.ocr.org.uk, choose your community and join the discussion!

Interchange

OCR Interchange has been developed to help you to carry out day to day administration functions online, quickly and easily. The site allows you to register and enter candidates online. In addition, you can gain immediate a free access to candidate information at you convenience. Sign up at https://interchange.ocr.org.uk

Published Resources

OCR offers centres a wealth of quality published support with a fantastic choice of 'Official Publisher Partner' and 'Approved Publication' resources, all endorsed by OCR for use with OCR specifications.

Publisher partners

OCR works in close collaboration with three Publisher Partners; Hodder, Heinemann and Oxford University Press (OUP) to ensure centres have access to:

- Better published support, available when you need it, tailored to OCR specifications
- Quality resources produced in consultation with OCR subject teams, which are linked to OCR's teacher support materials
- More resources for specifications with lower candidate entries
- Materials that are subject to a thorough quality assurance process to achieve endorsement

Hodder Education is the publisher partner for OCR GCE Computing

Oxford University Press (OUP) is producing the following resources for OCR GCE Computing for first teaching in September 2008, which will be available in Spring 2008.

OCR Computing for A level by Chris Leadbetter, Agneau, George Rouse

OCR Computing for A level Dynamic Learning Network Edition CD ROM by Chris Leadbetter, Agneau, George Rouse

Approved publications

OCR still endorses other publisher materials, which undergo a thorough quality assurance process to achieve endorsement. By offering a choice of endorsed materials, centres can be assured of quality support for all OCR qualifications.

Endorsement

OCR endorses a range of publisher materials to provide quality support for centres delivering its qualifications. You can be confident that materials branded with OCR's "Official Publishing Partner" or "Approved publication" logos have undergone a thorough quality assurance process to achieve endorsement. All responsibility for the content of the publisher's materials rests with the publisher.

These endorsements do not mean that the materials are the only suitable resources available or necessary to achieve an OCR qualification. Any resource lists which are produced by OCR shall include a range of appropriate texts.